

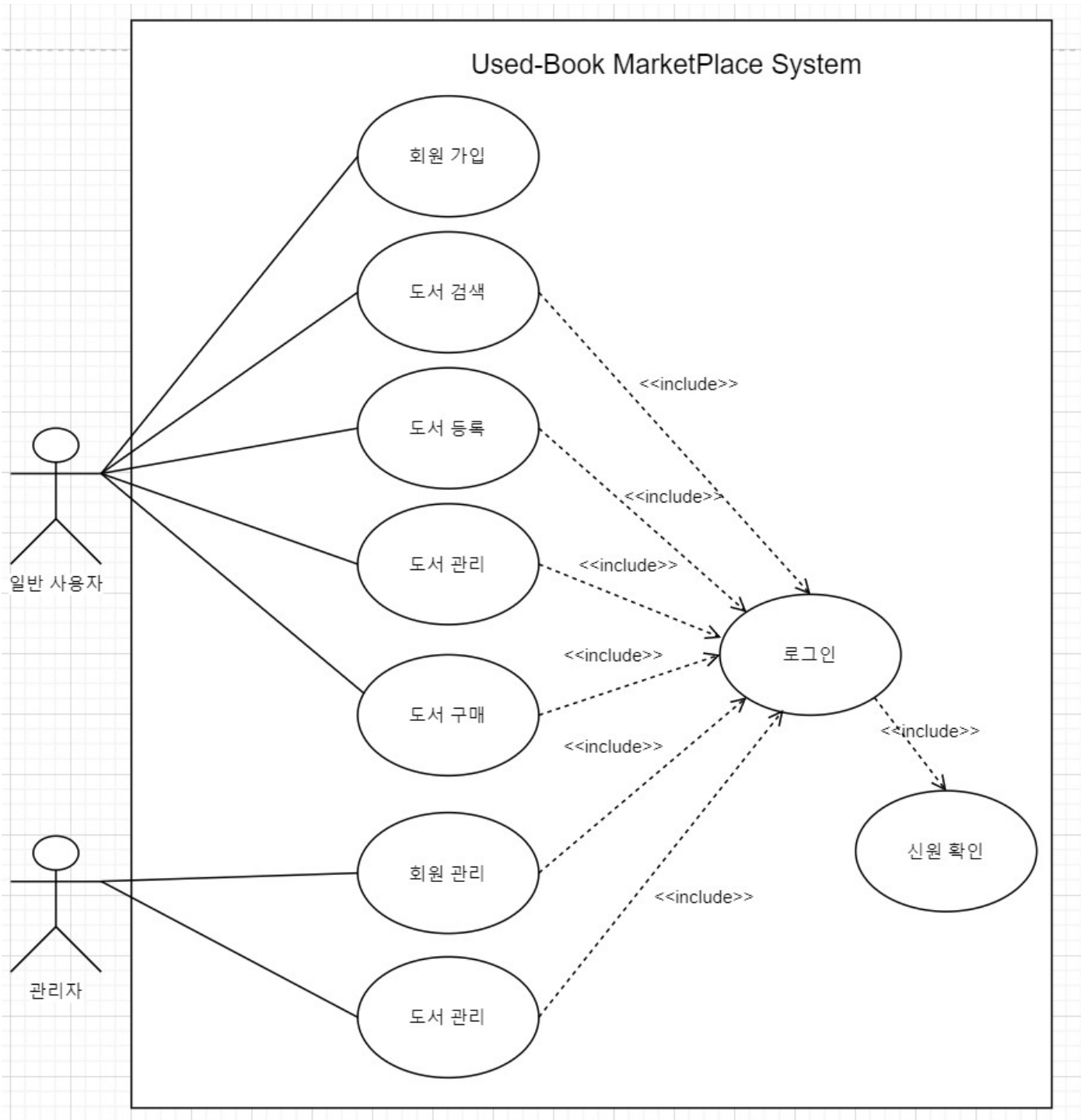
# 소프트웨어공학

-Term Project-



학 과 :	문헌정보학과
학 번 :	20142611
이 름 :	이하람
작성날짜 :	2020.06.25

## <유스케이스 다이어그램>



## <유스케이스 명세>

### 1) 사용자용

#### Use case UC1 : 도서 등록하기

- Scope : 헌책 장터 시스템
- Level : user goal
- Primary Actor : 일반 이용자
- Stakeholders and Interests :
  - ✓ 일반 이용자 : 도서를 팔기 위해 시스템상에 등록한다.
  - ✓ 관리자 : 등록된 도서의 적합성 여부를 판단하여 해당 도서를 삭제할 수 있다.
  - ✓ 시스템 : 이용자로부터 도서 정보를 입력 받아 저장한다.
- Preconditions : 기등록 회원이며, 해당 회원이 Activated한 상황이어야 한다.
- Success Guarantee : 도서가 성공적으로 시스템상에 등록이 되어 팔 수 있게 된다.
- Main Success Scenario
  1. 로그인을 한다.
  2. 사용자 메뉴를 보여준다.
  3. 도서 등록을 요청한다.
  4. 도서 정보를 입력한다.(도서이름은 필수)
    - 4.1 ISBN을 입력한다.(생략가능)
    - 4.2 저자를 입력한다. (생략가능)
    - 4.3 출판사를 입력한다. (생략가능)
    - 4.4 출판 년도를 입력한다. (생략가능)
    - 4.5 가격을 입력한다. (생략가능)
    - 4.6 상태를 입력한다. (생략가능)
  5. 시스템에 도서 정보를 저장한다.

- Alternative Flows

1a. 로그인 실패 시(아이디, 비밀번호오류, Deactivated)안내 메시지를 출력한다.

4. 도서 정보 등록 오류

4.1.a 문자 입력 시 다시 입력 받기.

4.4.a 문자 입력 시 다시 입력 받기.

4.5.a 문자 입력 시 다시 입력 받기.

4.6.a "Excellent", "Good", "Fair"이외 문자 입력 받을 시 다시 입력 받기.

### **Use case UC2 : 도서 구매하기**

- Scope : 헌책 장터 시스템

- Level : user goal

- Primary Actor : 일반 이용자

- Stakeholders and Interests :

✓ 일반 이용자 : 도서를 구매하기 위해 시스템에 로그인하며, 책을 검색 후 원하는 책을 구매한다(e-mail발송).

✓ 시스템 : 이용자의 요구에 따라 검색 된 책의 판매자 e-mail을 제공

- Preconditions :

✓ 판매자와 구매자가 기등록 회원이며, 모두 Activated한 상황이어야 한다.

✓ 시스템에 등록된 책이 최소 한권 이상은 있어야한다.

- Success Guarantee : 원하는 도서를 구매(판매자에게 e-mail발송)할 수 있게 된다.

- Main Success Scenario

1. 로그인을 한다.

2. 사용자 메뉴를 보여준다.

3. 도서 검색을 요청한다.
  4. 원하는 도서를 검색한다.(제목, ISBN, 저자, 출판사, 출판 년도, 판매자ID로 검색 가능)
  5. 시스템은 해당 도서의 User ID를 이용하여 판매자에게 구매 의사 이메일을 발송한다.
- Alternative Flows
    - 3a. 검색 내용에 해당하는 도서가 없으면 다시 사용자메뉴로 돌아 감.

## 2) 관리자용

### Use case UC3 : 일반 이용자 계정 삭제하기

- Scope : 현재 장터 시스템
- Level : user goal
- Primary Actor : 관리자
- Stakeholders and Interests :
  - ✓ 관리자 : 문제가 있는 이용자의 상태를 Deactivated로 변경한 뒤, 해당 유저를 시스템에서 삭제함.
  - ✓ 일반 이용자 : 관리자에 의해 Deactivated된 이용자는 로그인 불가. 또한, 삭제된 이용자는 시스템 상에 서 책을 포함한 모든 정보가 사라짐
  - ✓ 시스템 : 관리자의 요구에 따라 해당 이용자의 상태 변경 및 삭제 진행
- Preconditions :
  - ✓ 관리자는 시스템 상에 관리자로 등록되어 있음(회원 가입 불필요)
  - ✓ 시스템에 등록된 회원이 최소 한 명 이상이어야 한다.
- Success Guarantee : 이용자가 삭제되면 시스템 상의 해당 이용자의 모든 정보가 사라짐.
- Main Success Scenario
  1. 관리자 아이디로 로그인한다.
  2. 관리자 메뉴를 보여준다.
  3. 사용자 관리를 선택하면 시스템에 등록된 모든 사용자의 리스트를 보여준다.
  4. 사용자의 상태를 변경한다.(Activate -> Deactivate)

5. 사용자를 삭제한다.

- Alternative Flows

1a. 로그인 실패 시 안내 메시지 출력

3a. 삭제하려는 이용자가 이미 Deactivated인 경우

1. 바로 사용자를 삭제한다.

#### Use case UC4 : 관리자가 도서 삭제하기

- Scope : 헌책 장터 시스템

- Level : user goal

- Primary Actor : 관리자

- Stakeholders and Interests :

✓ 관리자 : 분쟁 등이 발생할 경우 문제의 도서를 시스템에서 삭제함

✓ 일반 이용자 : 삭제된 도서는 일반 이용자의 도서 리스트에서도 삭제됨

✓ 시스템 : 관리자의 요구에 따라 해당 도서를 삭제

- Preconditions :

✓ 관리자는 시스템 상에 관리자로 등록되어 있음(회원 가입 불필요)

✓ 시스템에 등록된 도서가 최소 한 권 이상이어야 한다.

✓ 관리자는 모든 책을 list up할 수 있어야한다.

- Success Guarantee : 도서가 삭제되면 시스템 상에서 해당 도서에 대한 모든 정보가 사라짐.(이용자 계정에서도 동일하게 삭제됨)

- Main Success Scenario

1. 관리자 아이디로 로그인한다.

2. 관리자 메뉴를 보여준다.

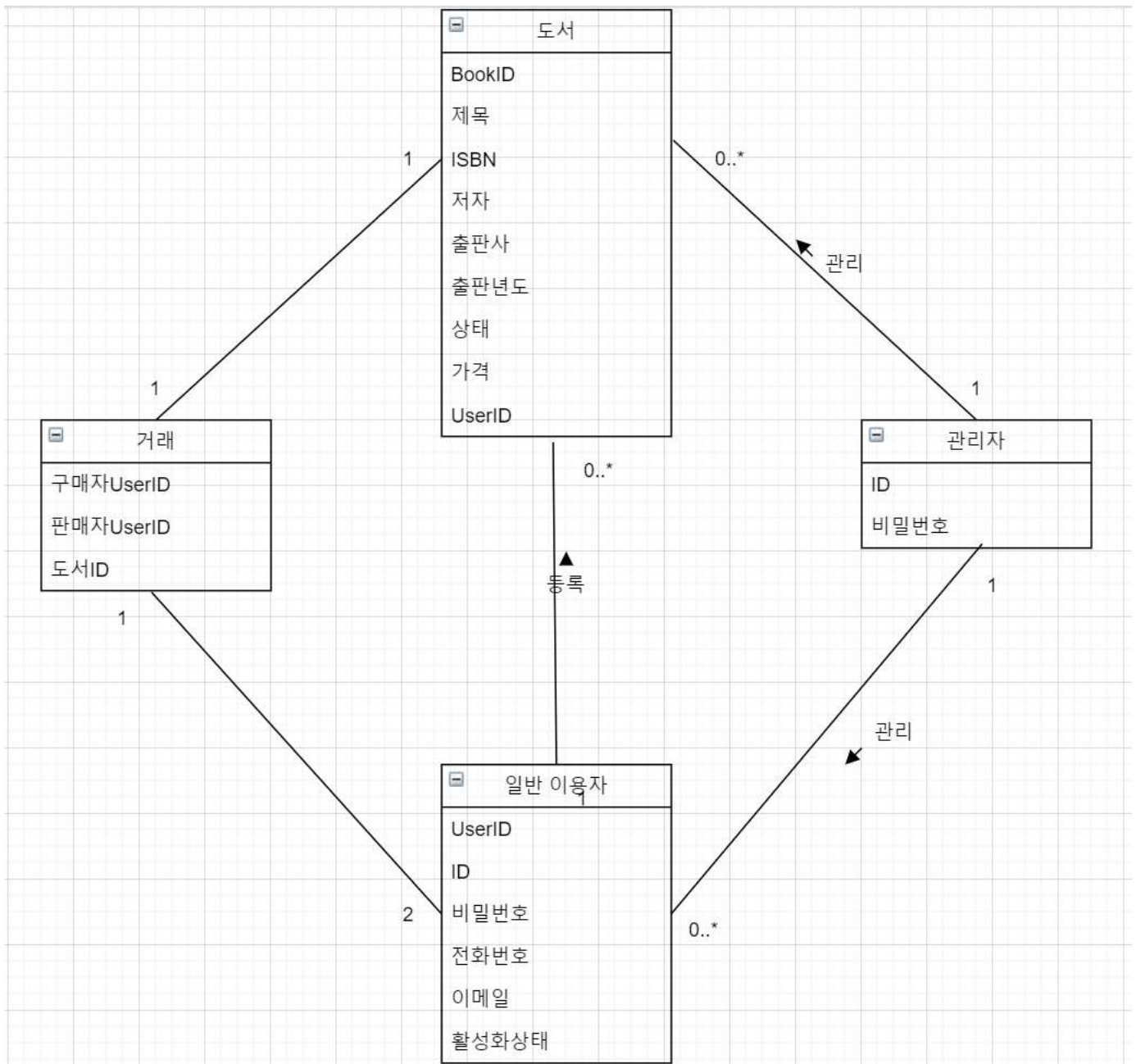
3. 도서 관리를 선택하면 시스템에 등록된 모든 도서의 리스트를 보여준다.

4. 도서를 삭제한다.

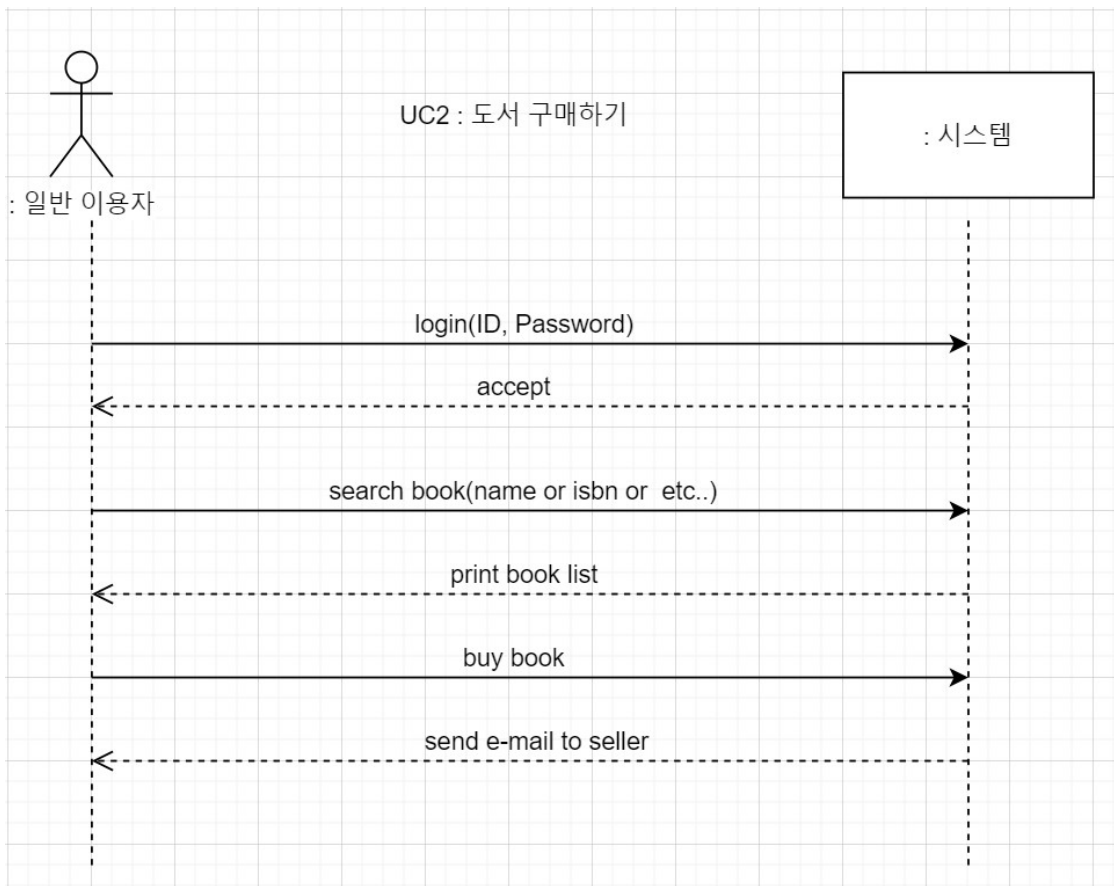
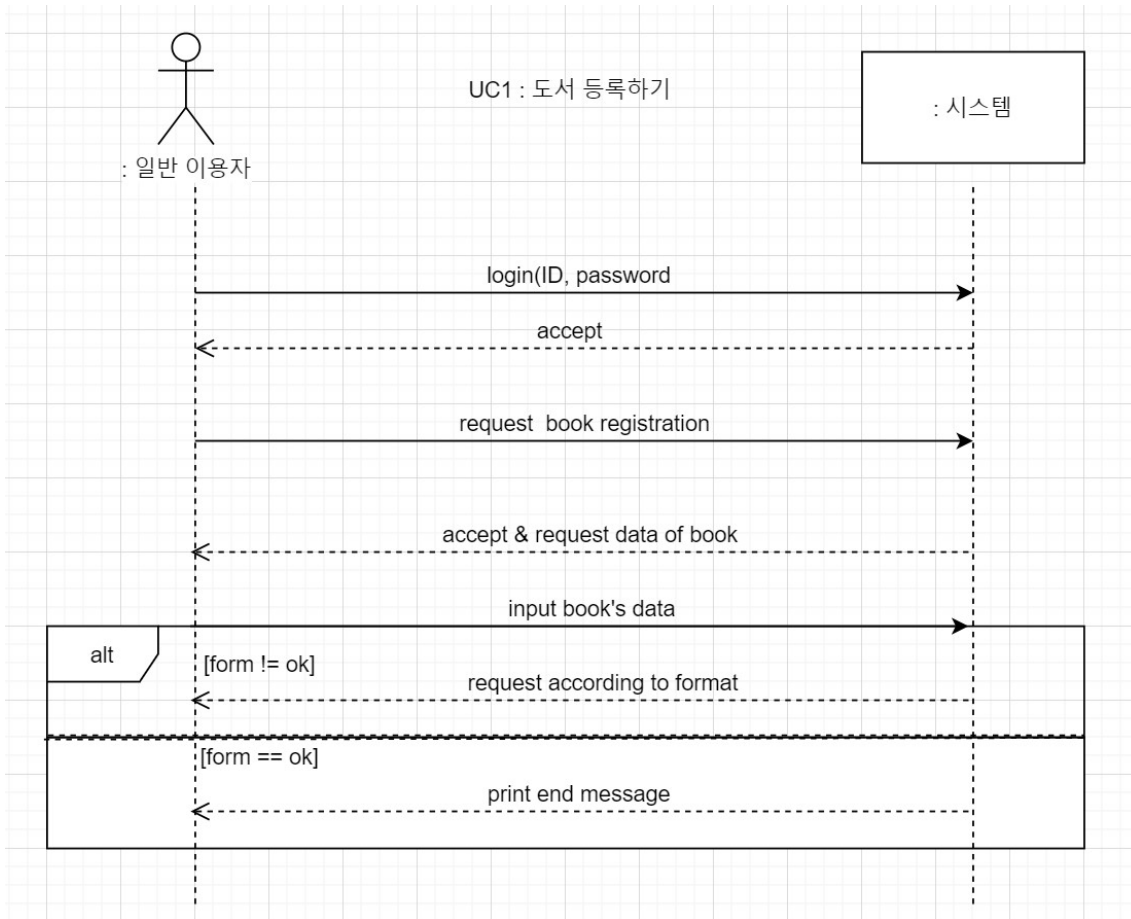
- Alternative Flows

1a. 로그인 실패 시 안내 메시지 출력

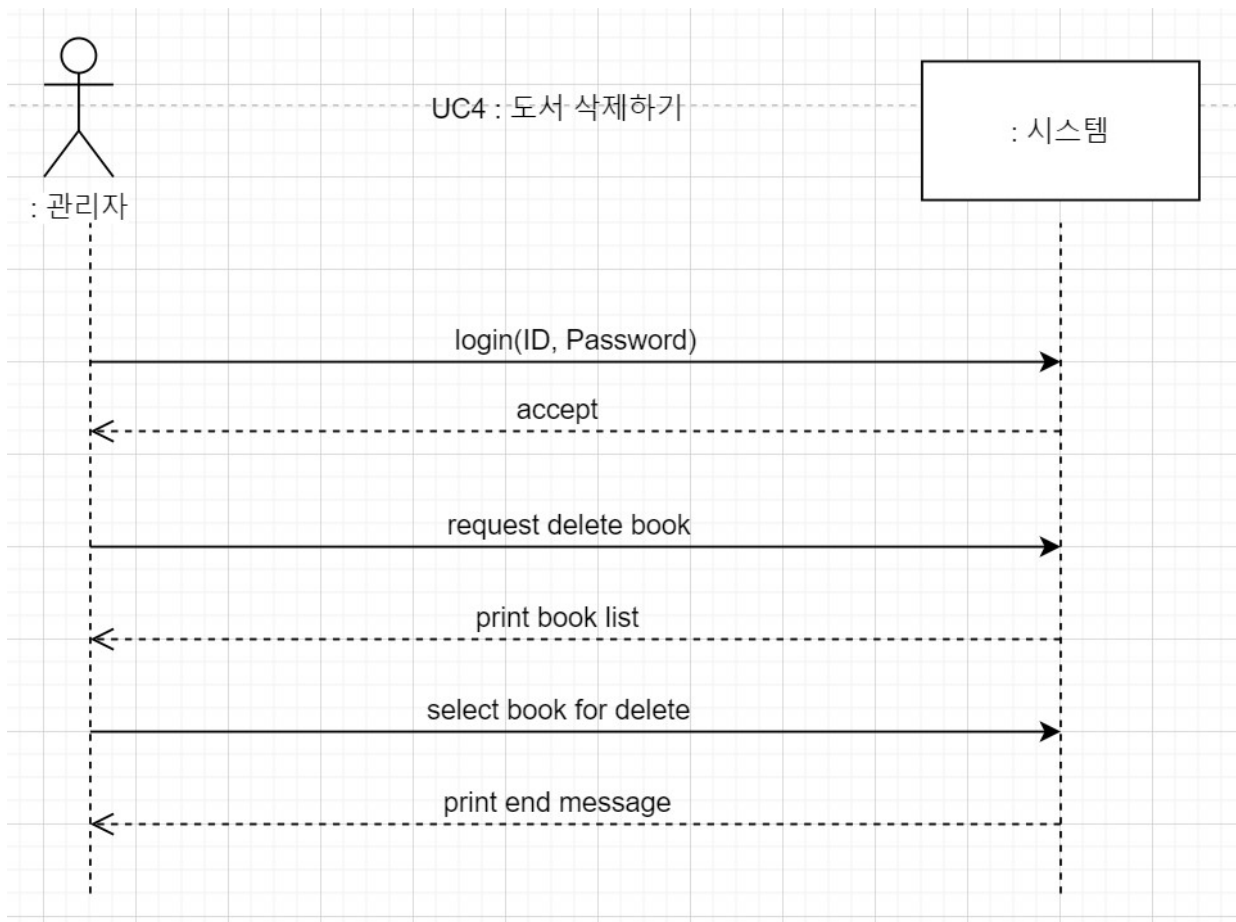
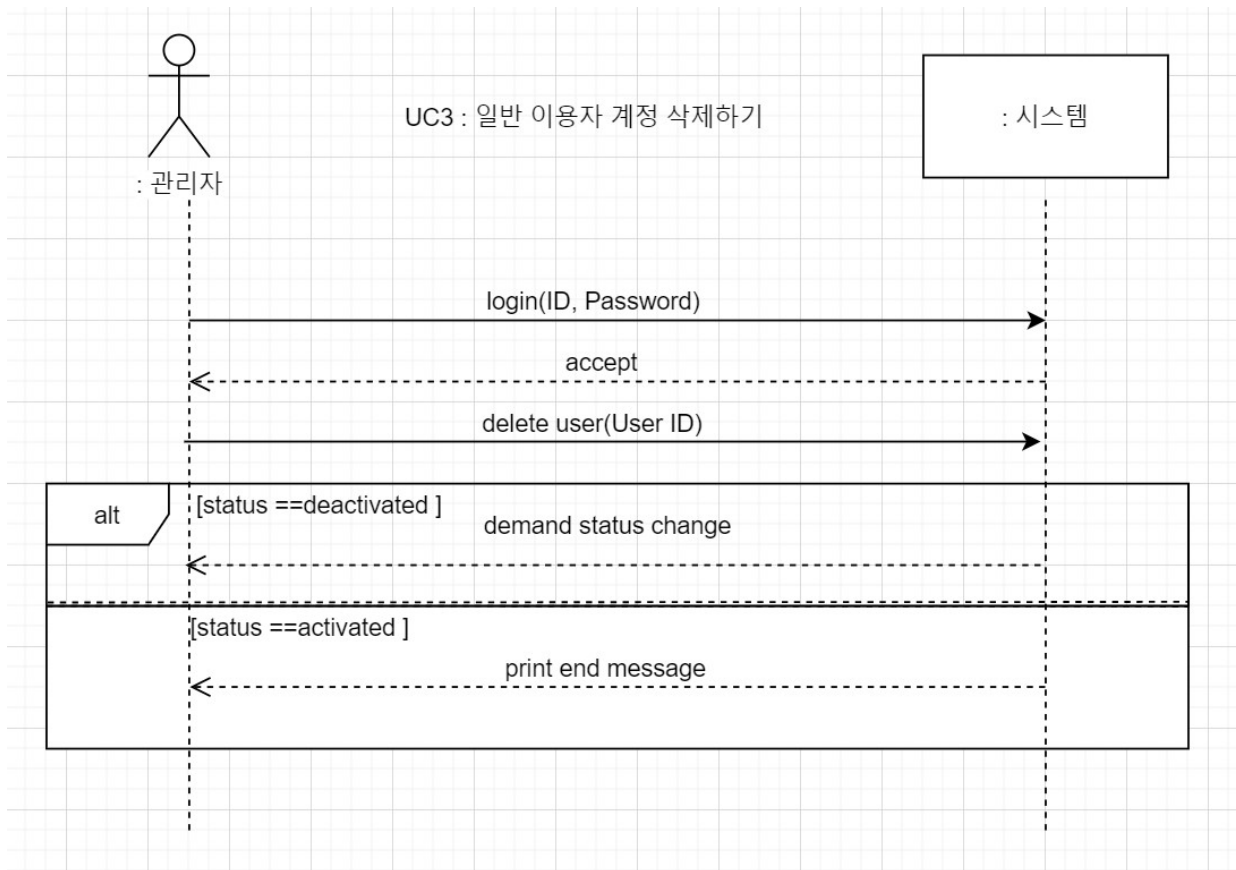
## <Domain Model>



## <System Sequence Diagram>







## <Operation Contract>

1)

Operation	login(ID : string, Password : string)
Cross References	All of use-case
Preconditions	The system user was already registered member
Postconditions	The user could use the system. For example, search the book, or buy the book.

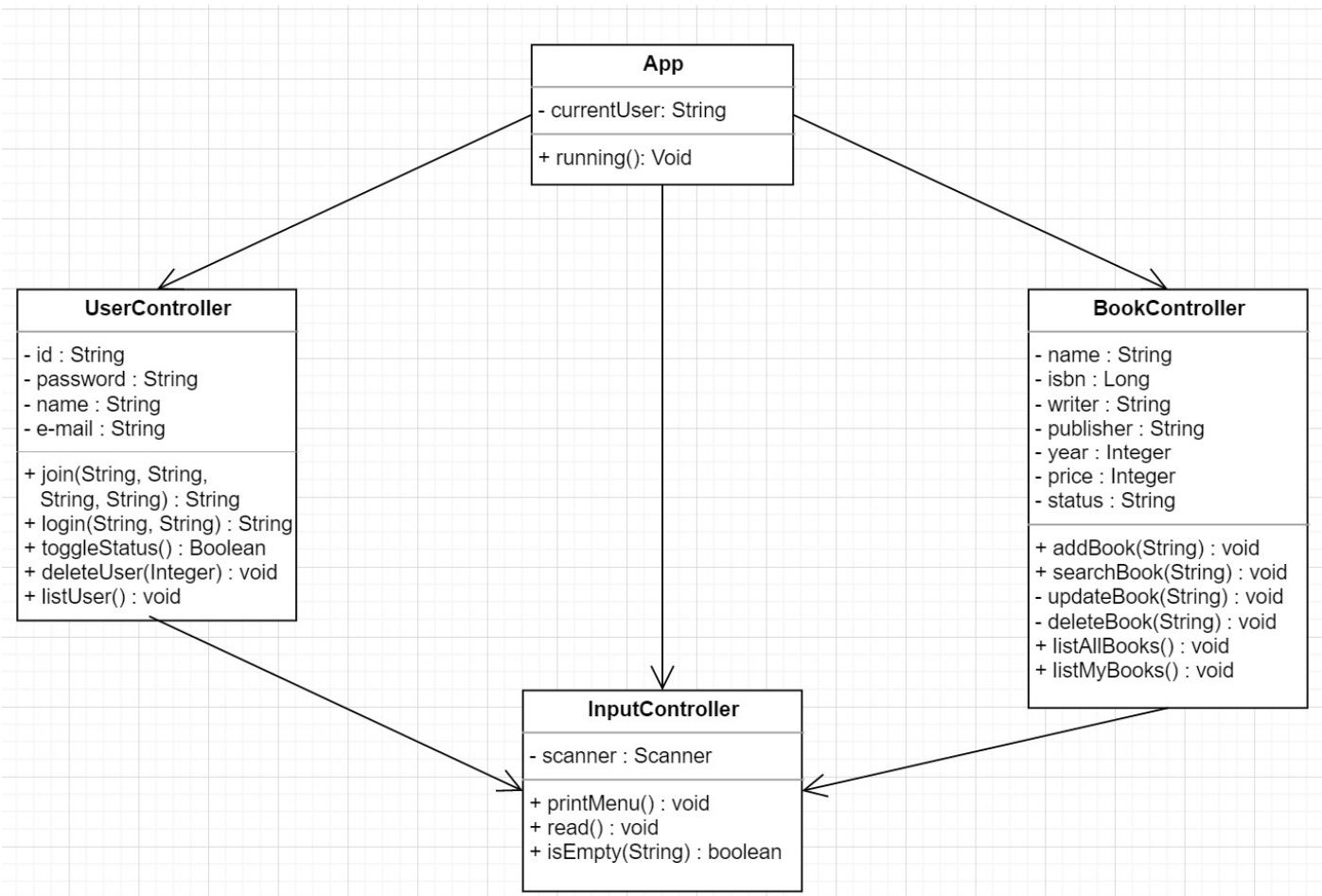
2)

Operation	Delete(UserID)
Cross References	UC3 : 일반 이용자 계정 삭제하기
Preconditions	At least one general user must be registered in the system.
Postconditions	- A User instance was deleted. - All book's instances registered by that user was be deleted from the system.

3)

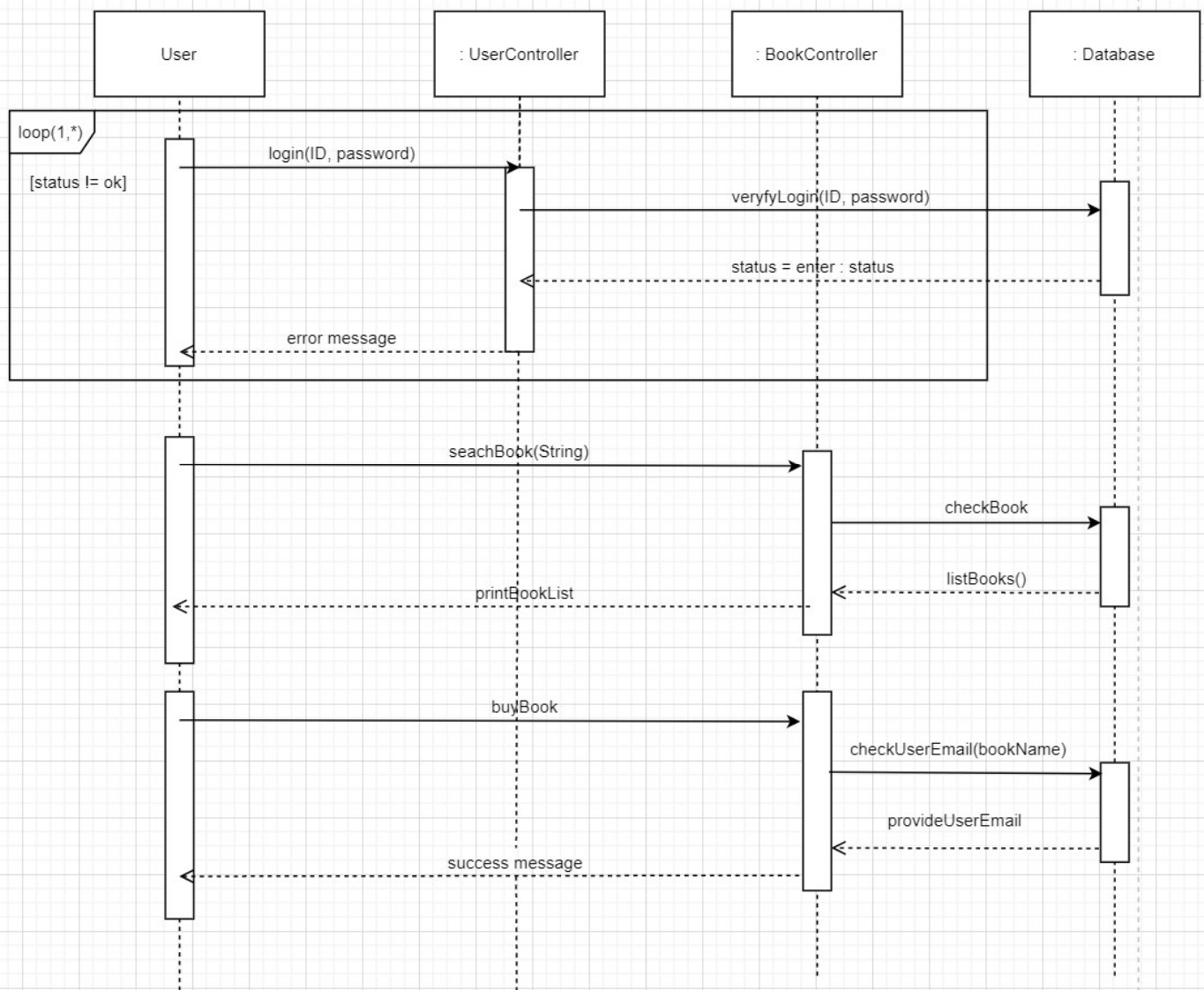
Operation	Delete(BookID)
Cross References	UC4 : 관리자가 도서 삭제하기
Preconditions	At least one book must be registered in the system.
Postconditions	A Book instance was deleted.

## <Design Class Diagram>

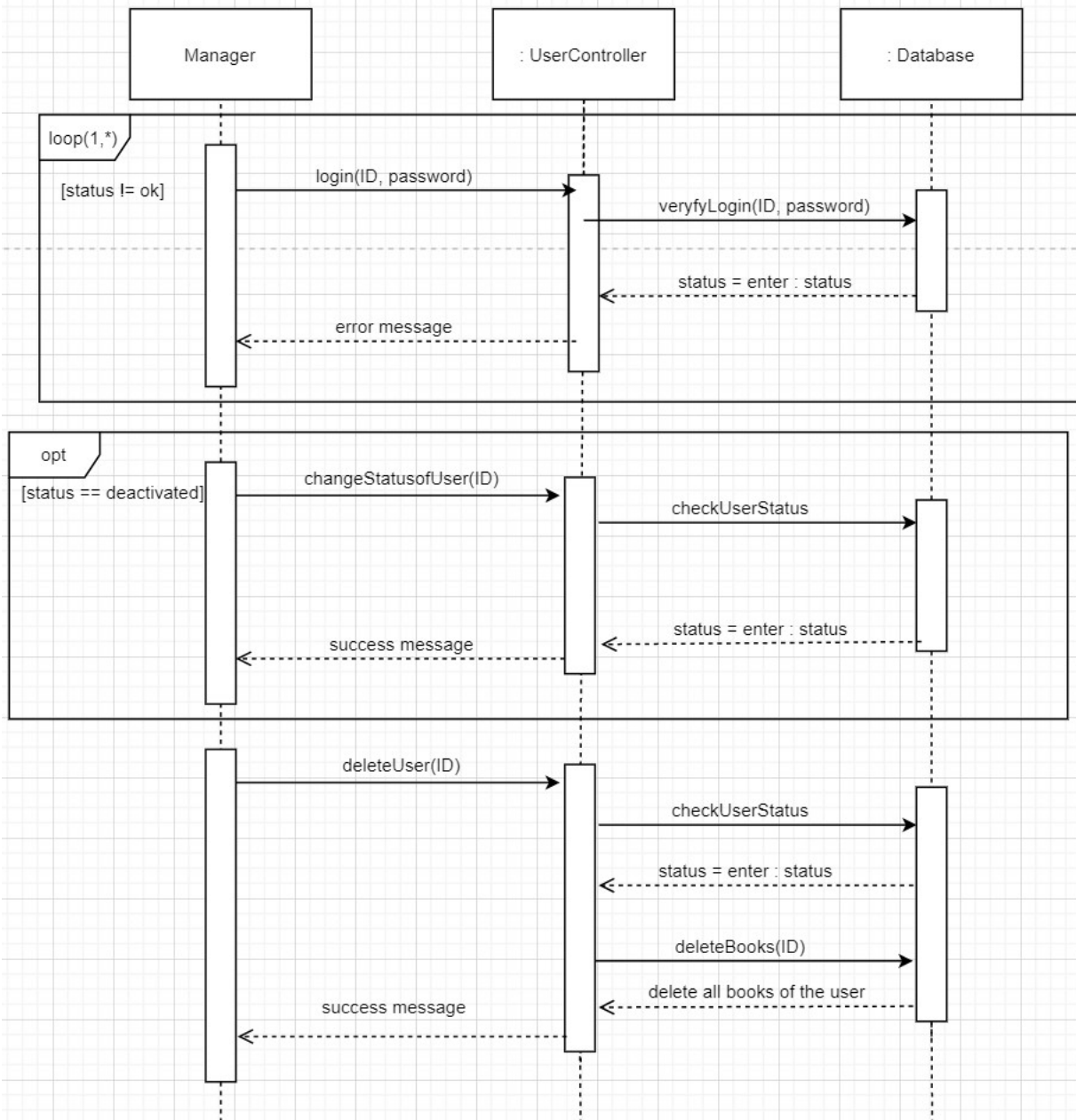


## <Sequence Diagram>

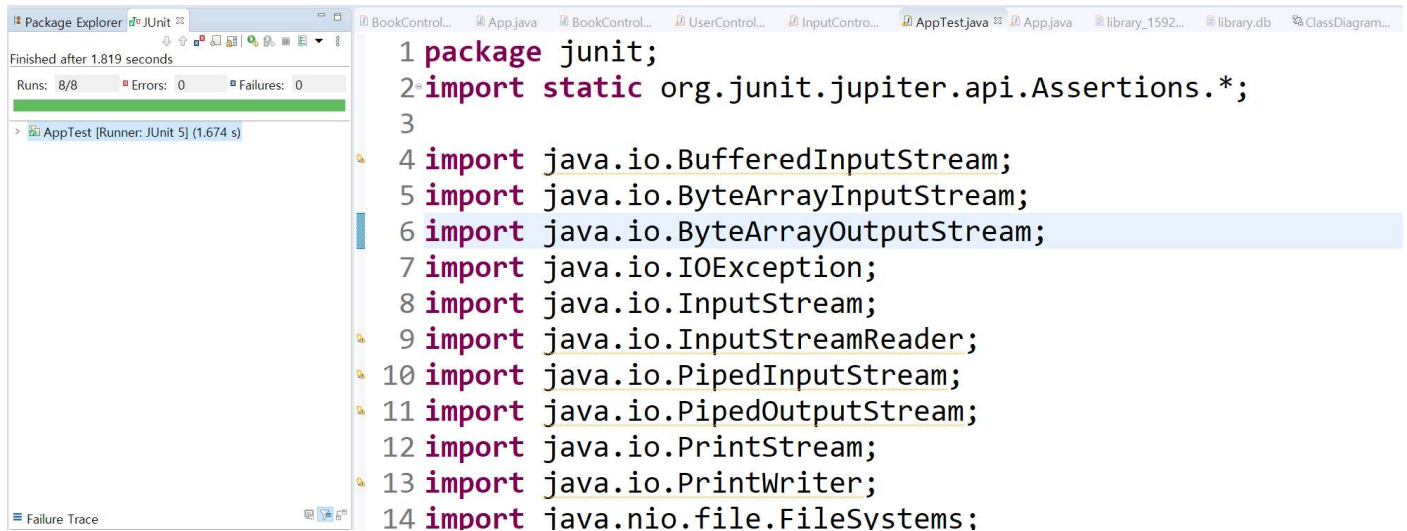
<도서 구매하기>



<일반 사용자 계정 삭제하기>



## JUnit 결과 : 성공률 100%



The screenshot shows an IDE interface. On the left, the 'Package Explorer' displays a project named 'JUnit'. Below it, a summary bar indicates 'Finished after 1.819 seconds', 'Runs: 8/8', 'Errors: 0', and 'Failures: 0'. A green progress bar is shown below this summary. Underneath, a list shows '> AppTest [Runner: JUnit 5] (1.674 s)'. At the bottom left, there is a 'Failure Trace' section. On the right, the 'AppTest.java' file is open, displaying the following Java code:

```
1 package junit;
2 import static org.junit.jupiter.api.Assertions.*;
3
4 import java.io.BufferedReader;
5 import java.io.ByteArrayInputStream;
6 import java.io.ByteArrayOutputStream;
7 import java.io.IOException;
8 import java.io.InputStream;
9 import java.io.InputStreamReader;
10 import java.io.PipedInputStream;
11 import java.io.PipedOutputStream;
12 import java.io.PrintStream;
13 import java.io.PrintWriter;
14 import java.nio.file.FileSystems;
```